Attorney Docket No.: 42.P17996

APPLICATION FOR UNITED STATES LETTERS OF PATENT

FOR

A METHOD FOR SIGNALING PCI/PCI-X STANDARD HOT-PLUG CONTROLLER (SHPC) COMMAND STATUS

Inventors: **Peter N. Martin; and**
Buck W. Gremel

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN, LLP
12400 Wilshire Boulevard, 7th Floor
Los Angeles, California 90025
(206) 292-8600

"Express Mail" Label Number EV320119974US

Date of Deposit December 31, 2003

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to New Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313

Jan Little-Washington

Jan Little-Washington

A METHOD FOR SIGNALING PCI/PCI-X STANDARD HOT-PLUG CONTROLLER (SHPC) COMMAND STATUS

BACKGROUND

1. Field

[0001] Embodiments of the present invention relate to hot-plug technology and, in particular, to peripheral component interconnect (PCI/PCI-X) standard hot-plug controllers (SHPCs).

2. Discussion of Related Art

[0002] A peripheral component interconnect (PCI/PCI-X) bus is a bus in a computer system that interconnects a microprocessor and peripheral devices, such as keyboards, disk drives, video adapters, etc. A PCI/PCI-X bus has slots into which the adapter cards for the peripheral devices can be inserted or removed. Hot-plug technology allows a user to physically remove or insert one or more PCI adapter cards without having to remove power to the entire system or re-booting the system software. Only the individual PCI/PCI-X slots are affected and the other devices in the system are not disrupted.

[0003] Hot-plug controllers were being developed by various vendors that were compatible with the *PCI Hot-Plug Specification, Revision 1.0*, October 6, 1997, PCI Special Interest Group, Portland, Oregon. It has been proposed that standardized hot-plug controllers be developed so that vendor-specific hot-plug controllers could be compatible across many platforms. The *PCI Standard Hot-Plug Controller and Subsystem Specification, Revision 1.0*, June 20, 2001, PCI Special Interest Group, Portland, Oregon, (hereinafter "SHPC Specification") was developed to meet this challenge.

[0004] The SHPC Specification provides that each slot for an adapter card have indicators, such as light emitting diodes (LEDs). For instance the SHPC Specification provides that there be two LEDs per slot, a power indicator (PLED) and an attention indicator (ALED). According to the SHPC Specification, each indicator is in one of three states: on, off, or blinking.

[0005] When the PLED is off, it is indicating that main power to the associated slot is off and that an adapter card may be safely inserted or removed from the slot. When the PLED is on, it is indicating that main power to the slot is on and that an adapter card may not be safely inserted or removed from the slot. When the PLED blinks, it is indicating that the slot is powering up or powering down and an adapter card cannot be safely inserted or removed from the slot.

[0006] When the ALED is on, it is indicating that an operational problem exists at the slot or adapter card. When the ALED is off, it is indicating that the slot and adapter card are operating normally. When the ALED is blinking, it is indicating that the system software is identifying the slot for a human operator to find. The SHPC Specification requires that when the PLED or ALED blinks, it blinks at a fifty percent duty cycle (±5%), such as 250 nanoseconds (ns) on and 250 ns off, at a prescribed frequency.

[0007] This indicator scheme has limitations, however, in that it does not communicate very much information about what the Standard Hot-Plug Controller (SHPC) is doing. The indicator scheme only shows whether a slot is active (the adapter card in the slot is powered), inactive (the adapter card in the slot is not powered), or in the process of changing states between being active and inactive, or vice versa.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally equivalent elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the reference number, in which:

[0009] Figure 1 is a high-level block diagram of a computer system according to an embodiment of the present invention;

[0010] Figure 2 is a high-level block diagram of the standard hot-plug controller depicted in Figure 1 according to an embodiment of the present invention;

[0011] Figure 3 is a flowchart illustrating a process for operating the standard hot-plug controller depicted in Figure 1according to an embodiment of the present invention;

[0012] Figures 4 through 15 illustrate blinking patterns according to an embodiment of the present invention; and

[0013] Figure 16 is a timing diagram illustrating a non-fifty percent blinking pattern for light emitting diode(s) on a PCI/PCI-X slot(s) according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS

[0014] Figure 1 is a high-level block diagram of a computer system 100 according to an embodiment of the present invention. The system 100 typically supports a high-performance desktop, a workstation, a server, etc. In one embodiment the system 100 is a UNIX platform. In other embodiments, the system 100 may be Windows® or Windows® NT platform. Those skilled in the art will appreciate that a variety of platforms may be used when implementing embodiments of the present invention.

[0015] The system 100 in the illustrated embodiment includes a microprocessor 102 coupled to a memory controller 104. The memory controller 104 is coupled to a bridge 106 and to memory 107. The bridge 106 may be coupled to the memory controller 104. The bridge 106 also is coupled to one or more peripheral component interconnect (PCI/PCI-X) slots represented by the slots 108, 110, 112, 114, 116, and 118 via a PCI/PCI-X bus 120.

[0016] The microprocessor 102 may be any suitable microprocessor that performs conventional functions of executing programming instructions including implementing many embodiments of the present invention. The microprocessor 102 can be a processor of the Pentium® processor family available from Intel Corporation of Santa Clara, California, but might be any processor that is capable of loading commands into the SHPC 128 or 130. In one embodiment, the microprocessor 102 includes software 122 that the microprocessor utilizes to load commands into the SHPC 128 or 130.

[0017] Of course, other software drivers independent of the microprocessor may be used to drive commands to the bridge 106. After reading the description herein, a person having ordinary skill in the relevant art will readily recognize how to implement embodiments of the present invention using other software drivers.

[0018] The illustrated memory 107 may be any suitable memory that performs its conventional functions of storing data (pixels, frames, audio, video, etc.) and software (control logic, instructions, code, computer programs, etc.) for access by other system 100 components. In general, the memory 107 includes several data lines corresponding to several addressable storage locations. The memory 107 may be any known dynamic random access memory (DRAM), static RAM (SRAM), Flash memory, etc. Memory technology is well known.

[0019] The memory controller 104 may be any suitable memory controller that performs conventional functions of controlling and monitoring the status of memory 107 data lines, error checking, etc. The memory controller 104 also may be a primary interface to the microprocessor 102 and the bridge 110. Memory controller technology is well known.

[0020] The bridge 106 interfaces the processor/memory subsystem (i.e., the microprocessor 102, memory controller 104, and memory 107) to the PCI/PCI-X bus 120 hierarchy. The bridge 106 includes a PCI-Express interface 123 that in the illustrated embodiment is coupled to two PCI-X interfaces 124 and 126.

Bridge technology is well known, PCI-Express technology is well known, and PCI-X technology is well known.

[0021] Although the bridge 106 is shown with the PCI-Express interface 123 coupled to the two PCI-X interfaces 124 and 126, embodiments of the present invention are not so limited. For example, the bridge 106 may be any host bridge, such as one capable of interfacing a processor/memory subsystem with a PCI/PCI-X bus. Alternatively, the bridge 106 may be any PCI/PCI-X-to-PCI/PCI-X bridge.

[0022] Each interface 124 and 126 is associated with a standard hot-plug controller (SHPC) 128 and 130, respectfully. Each SHPC 128 and 130 is associated with a PCI/PCI-X bus. In the illustrated embodiment, the SHPC 128 is associated with the PCI/PCI-X bus 120 and a PCI/PCI-X bus associated with the SHPC 130 is not shown. After reading the description herein, a person of ordinary skill in the relevant art will readily recognize how to implement the SHPC 130 for its associated PCI/PCI-X bus.

[0023] Each of the slots 108, 110, 112, 114, 116, and 118 is designed to accept an adapter card, which includes one or more PCI/PCI-X devices (e.g., printer, disk drive, keyboard, mouse, etc.) The SHPC 128 controls powering up and powering down of the slots 108, 110, 112, 114, 116, and 118 to allow adapter cards to be inserted or removed from the slots 108, 110, 112, 114, 116, and 118 without powering down the system 100 or re-booting the software 122.

[0024] Each SHPC is associated with slot control logic (not shown), which may be electronic component(s) that are responsible for providing signals to the SHPC 128. For example, the slot control logic may provide a signal to control the power state of one or more target PCI/PCI-X slots (e.g., PWREN). The slot control logic may provide a signal to control the connection of the PCI/PCI-X clock to one or more target PCI/PCI-X slots (e.g., CLKEN). The slot control logic may provide a signal to control the connection of various bus signals to one or more target PCI/PCI-X slots (e.g., BUSEN). The slot control logic may provide a signal to reset one or more target PCI/PCI-X slots (e.g., RST).

[0025] The SHPC Specification assumes that there are two light emitting diodes (LEDs) associated with and in close proximity to each slot 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, the slot 108 includes an LED 132 and an LED 134, the slot 110 includes an LED 136 and an LED 138, the slot 112 includes an LED 140 and an LED 142, the slot 114 includes an LED 144 and an LED 146, the slot 116 includes an LED 148 and an LED 150, and the slot 118 includes an LED 152 and an LED 154.

[0026] According to the SHPC Specification, the LEDs can be on, off, or blinking at a fifty percent duty cycle and the SHPC Specification explicitly states that an SHPC can never change the state of an indicator in response to an event such as a power fault.

[0027] According to embodiments of the present invention, the LEDs have blinking patterns that have a duty cycle or a frequency that is not the prescribed duty cycle or frequencyto communicate to an operator more information than whether the slot is active, inactive, or in the process of changing states between being active and inactive, or vice versa. For example, the LEDs may blink in a blinking pattern to communicate to an operator a particular command being processed, whether the command was processed successfully, whether power was applied to the slot if the command was not processed successfully, if the command was processed successfully, whether a fault occurred from which the bridge 106 is unlikely to recover and subsequent errant behavior is not only likely but is unpredictable (or a "hard error"), if the command was processed successfully, whether a fault occurred from which the bridge 106 is likely to be able to recover but the bridge 106 is being used in a manner that was not expected for normal operation (or a "soft error"), and/or whether power was applied to the target slots when the command was not processed successfully.

[0028] Figure 2 is a high-level block diagram of the SHPC 128 according to an embodiment of the present invention. The SHPC 128 in the illustrated embodiment includes a command register 202. The command register 202 is coupled to a hard error register 204, a soft error register 206, and a PCI/PCI-X slot(s) LED(s) blinking pattern controller 208. The hard error register 204 and soft error register 206 also are coupled to the blinking pattern controller 208.

[0029] Figure 3 is a flowchart illustrating a process 300 for operating the SHPC 128 according to an embodiment of the present invention. The operations of the process 300 are described as multiple discrete blocks performed in turn in a manner that is most helpful in understanding embodiments of the invention. However, the order in which they are described should not be construed to imply that these operations are necessarily order dependent or that the operations be performed in the order in which the blocks are presented.

[0030] Of course, the process 300 is only an example process and other processes may be used to implement embodiments of the present invention. A machine-accessible medium with machine-readable instructions thereon may be used to cause a machine (e.g., a processor) to perform the process 300.

[0031] In a block 302, a command 210 is loaded into the command register

202. Although depicted as a single register, the command register 202 may include one or more registers that store a command 210 received from the software 122.

[0032] One command may be an "LED-ON" command, which instructs the SHPC 128 to turn an LED for one or more target slots "on" solid. Another command may be an "LED-OFF" command, which instructs the SHPC 128 to turn an LED for one or more target slot "off." Another command may be a "LED-BLINK" command, which instructs the SHPC 128 to blink the LED for one or more target slots.

[0033] Another command may be a "PWRONLY" command, which instructs the SHPC 128 to power up one or more target slots without connecting clock or bus signals to the slots. Another command may be an "ENABLE" command, which instructs the SHPC 128 to power up one or more target slots, and to connect the clock and bus signals. Another command may be a "DISABLE" command, which instructs the SHPC 128 to disconnect power, clock, and bus signals from one or more target slots. Another command may be a "CHANGE PCI/PCI-X BUS SPEED" command, which instructs the SHPC 128 to change the speed of the PCI/PCI-X bus 120.

[0034] In a block 303, the process 300 determines whether execution of blinking patterns to communicate to an operator more information than whether the slot is active, inactive, or in the process of changing states between being active and inactive, or vice versa, is enabled. In one embodiment of the present invention, a one-bit field (ENABLE_FLASH_MODE bit) in debug space of the blinking pattern controller 208 is provided to indicate that the LEDs can execute the blinking patterns described herein. In this embodiment, the software 122 may poll this bit to determine whether it is set.

[0035] If the ENABLE_FLASH_MODE bit is not set, then in a block 305, the process 300 performs normal command processing. If the ENABLE_FLASH_MODE bit is set, then the process 300 passes to a block 304.

[0036] In the block 304, the process 300 determines whether the command 210 was successfully processed. If the command 210 was successfully processed, the process 300 executes a block 306.

[0037] In the block 306, the LED(s) on one or more target slots blinks in a blinking pattern associated with the particular command. In embodiments of the present invention, the blinking pattern controller 208 may be a register that is capable of being programmed to cause blinking patterns to vary based on a particular command, error, SHPC status, etc.

[0038] Figure 4 illustrates a blinking pattern for the commands to turn an LED on, off, or make the LED blink, according to embodiments of the present invention. For purposes of explanation, assume that the top LEDs are the power indicators (PLEDs) and the bottom LEDs are the attention indicators (ALEDs).

[0039] In one example, if the command is to turn a slot 108 LED "ON," then the ALED 134 blinks once, as indicated by the numeral "1" on the ALED 134 in Figure 4, to indicate that the command was successfully processed and the PLED 132 or the ALED 134 turns on solid as instructed by the command. The PLED 132 or the ALED 134 will continue to stay on until it is turned off. If the command is to turn a slot 108 LED "OFF," then the ALED 134 blinks once, as indicated by the numeral "1" on the ALED 134 in Figure 4, to indicate that the command was successfully processed and the PLED 132 or the ALED 134 turns off as instructed by the command.

[0040] If the command is to make a slot 108 LED blink, then the ALED 134 blinks once, as indicated by the numeral "1" on the ALED 134 in Figure 4 to indicate that the command was successfully processed and the PLED 132 or the ALED 134 blinks as instructed by the command. The PLED 132 or the ALED 134 will continue to blink for an undetermined amount of time until it is caused to stop.

[0041] Figure 5 illustrates a blinking pattern for a PWRONLY command for the slot 108. In the illustrated embodiment, the ALED 134 blinks twice, as indicated by the numerals "1" and "2" on the ALED 134 in Figure 5, to indicate that the PWRONLY command was successfully processed.

[0042] Figure 6 illustrates a blinking pattern for an ENABLE command for the slot 110. In the illustrated embodiment, the ALED 138 blinks three times, as indicated by the numerals "1" and "2" and "3" on the ALED 138 in Figure 6 to indicate that the ENABLE command was successfully processed.

[0043] Figure 7 illustrates a blinking pattern for a DISABLE command for the slot 116. In the illustrated embodiment, the ALED 150 blinks four times, as indicated by the numerals "1" and "2" and "3" and "4" on the ALED 150 in Figure 7to indicate that the DISABLE command was successfully processed.

[0044] Figure 8 illustrates a blinking pattern for a CHANGE PCI/PCI-X BUS SPEED command, which affects all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, each ALED 134, 138, 142, 146, 150, and 154 blinks five times, as indicated by the numerals "1" and "2" and "3" and "4" and "5" on the ALEDs 134, 138, 142, 146, 150, and 154 in Figure 8to indicate that the CHANGE PCI/PCI-X BUS SPEED command was successfully processed.

[0045] Figure 9 illustrates a blinking pattern for a PWRONLY-ALL command, which affects all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, each ALED 134, 138, 142, 146, 150, and 154 blinks twice, as indicated by the numerals "1" and "2" on the ALEDs 134, 138, 142, 146, 150, and 154 in Figure 9 to indicate that the PWRONLY-ALL command was successfully processed.

[0046] Figure 10 illustrates a blinking pattern for an ENABLE-ALL command, which affects all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, each ALED 134, 138, 142, 146, 150, and 154 blinks three times, as indicated by the numerals "1" and "2" and "3" on the ALEDs 134, 138, 142, 146, 150, and 154 in Figure 10 to indicate that the ENABLE-ALL command was successfully processed.

[0047] Returning back to Figure 3, after the LED(s) on one or more target slots blinks in a blinking pattern associated with the particular command to indicate that the command has been processed successfully, a block 307 determines whether an error occurred. If an error has not occurred, then control returns to the block 302. If an error has occurred, a block 308 determines whether a soft error has occurred. The soft error register 206 may be a register that stores an indication that a soft error occurred. An example of a soft error may be an attempt to load the command register 202 while a command 210 is being processed. If a soft error has occurred, then a block 310 executes a "soft error" blinking pattern.

[0048] Figure 11 illustrates a "soft error" blinking pattern according to an embodiment of the present invention. In the illustrated embodiment, the ALEDs 134, 138, 142, 146, 150, and 154 blink once in consecutive order, with the ALED 154 blinking first as indicated by the numeral "1" on the ALED 154, the ALDE 150 blinking second as indicated by the numeral "2" on the ALED 150, the ALED 146 blinking third as indicated by the numeral "3" on the ALED 146, the ALED 142 blinking fourth as indicated by the numeral "4" on the ALED 142, the ALED 138 blinking fifth as indicated by the numeral "5" on the ALED 138, and the ALED 134 blinking sixth as indicated by the numeral "6" on the ALED 134, , with the first, the ALED 150 blinks once second,

[0049] If an error has occurred, after the LED(s) on one or more target slots blinks in a blinking pattern associated with the particular command to indicate that the command has been processed successfully, the a block 312 determines whether a hard error has occurred. The hard error register 204 may be a register that stores an indication that a hard error has occurred. An example of a hard error may be loading of a "Reserved" command into the command register 202. If a hard error has occurred, then a block 310 executes a "hard error" blinking pattern.

[0050] Figure 12 illustrates a "hard error" blinking pattern according to an embodiment of the present invention. In the illustrated embodiment, the ALEDs 134, 138, 142, 146, 150, and 154 blink once in consecutive order, with the ALED 154 blinking sixth as indicated by the numeral "6" on the ALED 154, the ALED 150 blinking fifth as indicated by the numeral "5" on the ALED 150, the ALED 146 blinking fourth as indicated by the numeral "4" on the ALED 146, the ALED 142 blinking third as indicated by the numeral "3" on the ALED 142, the ALED 138 blinks second as indicated by the numeral "2" on the ALED 138, and the ALED 134 blinking first as indicated by the numeral "1" on the ALED 134

[0051] Returning back to Figure 3, if in the block 304 it is determined that the command was not successfully processed, the process 300 executes a block 316. In the block 316, the process 300 determines whether power is applied to the target slot(s). If power is not applied to the target slot(s), a block 318 executes a blinking pattern that indicates that the command was not successfully processed and that power is not applied to the target slot(s).

[0052] Figure 13 illustrates a blinking pattern according to an embodiment of the present invention that is executed when the command 210 loaded into the command register 202 is "PWRONLY-ALL," the command 210 was not successfully processed, and power was not applied to all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment each PLED 132, 136, 140, 144, 148, and 152 blinks twice, as indicated by the numerals "1" and "2" on the PLEDs 132, 136, 140, 144, 148, and 152 in Figure 13 to indicate that power has not been applied to the slots 108, 110, 112, 114, 116, and 118. To sum up, if the command 212 is PWRONLY-ALL, the command 212 is not successfully processed, and power is not applied to the slots 108, 110, 112, 114, 116, and 118, then each ALED 134, 138, 142, 146, 150, and 154 blinks twice (Figure 9) and each PLED 132, 136, 140, 144, 148, and 152 blinks twice (Figure 13)..

[0053] Figure 14 illustrates a blinking pattern according to an embodiment of the present invention that is executed when the command 210 loaded into the command register 202 is "ENABLE-ALL" and the command 210 was not successfully processed because power was not applied to all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, each PLED 132, 136, 140, 144, 148, and 152 blinks three times, as indicated by the numerals "1" and "2" and "3" on the PLEDs 132, 136, 140, 144, 148, and 152 in Figure 14 to indicate that the ENABLE command was not successfully processed and power has not been applied to the slots 108, 110, 112, 114, 116, and 118. .

[0054] For example, each PLED 132, 136, 140, 144, 148, and 152 blinks three times, as indicated by the numerals "1" and "2" and "3" on the PLEDs 132, 136, 140, 144, 148, and 152 in Figure 14to indicate that the ENABLE command was not successfully processed and power has not been applied to the slots 108, 110, 112, 114, 116, and 118. To sum up, if the command 212 is ENABLE-ALL, the command 212 is not successfully processed, and power is not applied to the slots 108, 110, 112, 114, 116, and 118, then each PLED 132, 136, 140, 144, 148, and 152 blinks three times (Figure 14).

[0055] Referring back to Figure 3, if in the block 316, the process 300 determines that power is applied to the target slot(s), the process 300 executes a block 320. The block 320 executes a blinking pattern that indicates that power is applied to the target slot(s).

[0056] Figure 15 illustrates a blinking pattern according to an embodiment of the present invention that is executed when the command 210 loaded into the command register 202 is "PWRONLY-ALL," the command 210 was not successfully processed, and power was applied to all slots 108, 110, 112, 114, 116, and 118. In the illustrated embodiment, each PLED 132, 136, 140, 144, 148, and 152 and each ALED 134, 138, 142, 146, 150, and 154 blinks twice, as indicated by the numerals "1" and "2" on the PLEDs 132, 136, 140, 144, 148, and 152 in Figure 13 to indicate that the command 210 loaded into the command register 202 is "PWRONLY-ALL," the command 210 was not successfully processed, and power was applied to all slots 108, 110, 112, 114, 116, and 118

[0057] Of course, other blinking patterns may be used to communicate to an operator a particular command being executed, whether the command was processed successful, whether an error occurred after the command was executed and/or whether the error is a "hard" error or a "soft" error, whether power is applied to the target slot(s), and/or whether power was applied to the target slots when the command was not processed successfully. After reading the description herein, a person of ordinary skill in the relevant art will readily recognize how to implement embodiments of the present invention using other blinking patterns.

[0058] Recall that in embodiments of the present invention, the LEDs have blinking patterns that are less than or greater than a fifty percent duty cycle (±5%) to communicate to an operator information more than whether the slot is active, inactive, or in the process of changing states between being active and inactive, or vice versa. Figure 16 is a timing diagram 1600 for a blinking pattern having a duty cycle that is less than or greater than a fifty percent (±5%).

[0059] In the illustrated embodiment, the cadence of the blinking pattern has an "on" period 1602 followed by an "off" period 1604 followed by another "off" period 1606 that marks the end of a command 210 blinking pattern and the start of a hard error blinking pattern and/or a soft error blinking pattern. Thus, if there is no hard error or soft error after the command 210 has been executed, then the "off" period 1606 may last until the next command 210 is executed. In one embodiment, the "on" period 1602 is one hundred twenty-eight milliseconds (128ms), the "off" period 1604 is 128ms, and the "off" period 1606 is 500ms.

[0060] In embodiments of the present invention, the blinking pattern controller 208 may be a low priority requester of common resources in the system 100. Therefore, loading of a command 210 into the command register 202 during execution of a blinking pattern may cause the blinking pattern to be aborted and control may return to the SHPC 128. In these embodiments, the blinking pattern controller 208 may not flush the pending blinking pattern execution signal 212. Instead, the blinking pattern controller 208 may wait for the command 210 to be loaded before retrying the blinking pattern.

[0061] In one embodiment of the present invention, a one-bit field in debug space of the blinking pattern controller 208 is provided to indicate that a blinking pattern is being executed. In this embodiment, the software 122 may poll this bit to determine when to load the next command 210 to ensure that a pending blinking pattern is not squashed. The debug space also may be used to change the duty cycle of the blinking pattern.

[0062] Embodiments of the present invention may be implemented using hardware, software, or a combination thereof. In implementations using software, the software may be stored on a machine-accessible medium.

[0063] A machine-accessible medium includes any mechanism that provides (i.e., stores and/or transmits) information in a form accessible by a machine (e.g., a computer, network device, personal digital assistant, manufacturing tool, any device with a set of one or more processors, etc.). For example, a machine-accessible medium includes recordable and non-recordable media (e.g., read only memory (ROM), random access memory (RAM), magnetic disk storage media, optical storage media, flash memory devices, etc.), as well as electrical, optical, acoustic, or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.).

[0064] In the above description, numerous specific details, such as particular processes, materials, devices, and so forth, are presented to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that embodiments of the present invention can be practiced without one or more of the specific details, or with other methods, components, etc. In other instances, well-known structures or operations are not shown or described in detail to avoid obscuring the understanding of this description.

[0065] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, process, block, or characteristic described in connection with an embodiment is included in at least one embodiment of the present invention. Thus, the appearance of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification does not necessarily mean that the phrases all refer to the same embodiment. The particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0066] The terms used in the following claims should not be construed to limit embodiments of the invention to the specific embodiments disclosed in the specification and the claims. Rather, the scope of embodiments of the invention is to be determined entirely by the following claims, which are to be construed in accordance with established doctrines of claim interpretation.